

# How to Create, Modify, and Interface Aspen In-House and User Databanks for System Configuration 1:

*D. W. Camp*

**October 27, 2000**

**U.S. Department of Energy**

Lawrence  
Livermore  
National  
Laboratory

## **DISCLAIMER**

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U. S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

**How To Create, Modify, And Interface  
Aspen In-House and User Databanks  
for  
System Configuration 1:**

**Aspen Engine is on a UNIX Solaris server  
User's Aspen GUI is on the User's Windows NT PC  
User's Word is on the User's Windows NT PC  
User works from his/her own Windows NT PC**

**Revision 1**

**David W. Camp**

**October 27, 2000**

## Contents

<b>1. Introduction</b>	<b>5</b>
<b>2. System Configuration: Description of Computers and Directory Structure</b>	<b>6</b>
<b>3. General Comments and Strategies on User and In-House Databanks</b>	<b>8</b>
<b>4. Mechanics of Creating and Modifying Databank Files</b>	<b>9</b>
4.1. Introduction	9
4.2. Step by Step Instructions	10
<b>5. Mechanics of Telling Your User Interface about the Databank File</b>	<b>12</b>
5.1. Introduction	12
5.2. Step by Step Instructions	12
<b>6. Testing and Using User and In-House Databanks in Simulations</b>	<b>16</b>
<b>7. Miscellaneous Notes on Creating and Modifying Databank Files</b>	<b>19</b>
7.1. Names, passwords, file names, extensions, and case for databanks, etc.	19
7.2. Copying from one Databank to Another	21
7.3. Can't Change Component Names Easily	21
7.4. Limitations on Number of Parameters Allowed in a Databank	22
7.5. Documenting the source of a parameter value – the "srcode" parameter	22

## Appendices

<b>A. Notes and Examples for dfms Input Files and User Interface Databank Input Files</b>	<b>23</b>
<b>Part 1. dfms Input Files</b>	<b>23</b>
A.1 Summary:	23
A.2 Character and line formatting.	23
A.3 Making Multiple Databank Changes in One Input File	23
A.4 Examples	24
A.4.1 Create new db, add components, add parameters, enter values using COMP-LIST	24
A.4.2 Check the Contents of a Databank File	25
A.4.3 Add components, add parameters, enter values to existing databank, copy from databank	26
A.4.4 Create new, empty databank	26
A.4.5 Copy all data on some components from a User databank and copy some data on some components from a System databank to an In-House databank	27
Copy some parameters of some components from PURE10 to In-House dbank	27
A.4.6 Delete components from databank and change values of certain parameter elements	27
A.4.7 Copy all data from all components from a User to an In-House databank	29
<b>Part 2. User Interface Databank Input Files</b>	<b>32</b>
A.5 User Interface Databank Input Files	32
A.5.1 Tell GUI about databank components added by dfms in Example A.4.1	32
A.5.2 Not applicable	32
A.5.3 Update GUI to include new components added in Example A.4.3	32
A.5.4 Not necessary	33
A.5.5 Tell the GUI about In-House databank components of Ex. A.4.4 and A.4.5	33
A.5.6 Update the GUI to include components added to USRPP1B in Ex. A.4.6	34
A.5.7 Tell the GUI about In-House components from Ex. A.4.4, A.4.5, and A.4.7	34

<b>B. Toolkit of Computer Skills for Databank Work</b>	<b>35</b>
B.1 Type the UPPER or lower case accurately	35
B.2 A minimal ability to work in DOS and UNIX is needed	35
B.3 Check that directories are mapped and accessible as needed	35
B.4 Log in to <b>enginehost</b> from <b>UserNT</b> and open a UNIX command window there	36
B.5 Get write permission in the Aspen engine directory	36
B.6 Set XDATA environment	36
B.7 Put a text file onto <b>enginehost</b> that can be read by dfms, such as a <b>dbinname.inp</b> file	36
B.8 Convert a text file from DOS format to UNIX format using dos2unix	37
B.9 To open and view a dfms report file, <b>inrpname.rep</b> , using Word	37
B.10 To open a DOS command window on <b>UserNT</b> and get to a certain directory	37
B.11 Start up Aspen from your GUI	38
B.12 [Not Applicable to System Configuration 1]	38
B.13 [Not Applicable to System Configuration 1]	39
B.14 Prepare a text file on Word, such as <b>uifilename.dat</b> , and save onto directory <b>\custom</b>	40
<b>C. Some DOS Commands</b>	<b>41</b>
<b>D. Some UNIX Commands</b>	<b>42</b>
<b>E. Basic Text Editing in UNIX vi:</b>	<b>44</b>
<b>F. Questions to Resolve</b>	<b>45</b>
F.1 Properties used in simulations	45
F.2 DESCRIPTION sentence single quotes	45
F.3 Multiple User databanks	45
F.4 Using different directories	45
F.5 When copying from PURE10, limit of 40 parameters is exceeded	45
F.6 setenv XDATA needed despite instructions to contrary	45
F.7 Changing Component Name and Alias	46
F.8 Overlapping component names and databank contents	46
<b>G. Your Particular Computer Names, Directories, Addresses, and Paths</b>	<b>48</b>

## 1. Introduction

**The goal of this document is to provide detailed instructions to create, modify, interface, and test Aspen User and In-House databanks with minimal frustration.** The level of instructions are aimed at a novice Aspen Plus simulation user who is neither a programming nor computer-system expert.

The instructions are tailored to Version 10.1 of Aspen Plus and the specific computing configuration summarized in the Title of this document and detailed in Section 2. Many details of setting up databanks depend on the computing environment specifics, such as the machines, operating systems, command languages, directory structures, inter-computer communications software, the version of the Aspen Engine and Graphical User Interface (GUI), and the directory structure of how these were installed. Chapter 2 and Appendix G define these.

The following Aspen references are necessary for learning about Aspen databanks and for following and extending beyond this guide. Other Aspen User and Reference Manuals on physical properties provide background and guidance on using databanks.

- Aspen Plus Reference Manual: "Physical Property Data", Version 10. pp. 1-1 to 1-16.
- Aspen Plus Reference Manual: "System Management", Version 10, pp. 6-1 to 6-18
- Aspen document "Configuring Physical Property Databanks for ASPEN PLUS Release 10.1 (NT/Windows 95)", available from Aspen's on-line support library.

These Aspen references are necessary but not sufficient to get the job done efficiently without much frustration. The level of detail is insufficient for a nonexpert, some instructions and examples did not work in our specific environment, and many ambiguities and error-producing examples were only resolved by trial and error.

This document's instructions are intended to provide enough detail and correctness for this specific computing configuration to avoid problems with items such as the UPPER/lower/Title case of file names, file extensions, databank names, and commands, directory structures, command-line environments, terminal communications, and text character incompatibilities.

For a **tutorial** that goes through all the steps in detail, follow the exercise suggested in the last paragraph of Section 4.1.

You will need to have a minimum "toolkit" of computer skills to be successful. All such skills and tasks are described in Appendices B-E in detail that should be sufficient for the novice.

This report only deals with pure-component databanks. Creating, modifying, and interfacing binary databanks appears to be analogous, but a bit trickier.

A sister document, "How To Create, Modify, And Interface Aspen In-House and User Databanks for System Configuration 2: Aspen Engine is on a UNIX Solaris server; Group's Aspen GUI is on a Windows NT server; Group's Word is on a Windows NT server; User works from a diskless Sun workstation" provides instructions for this other computing environment.

## **2. System Configuration: Description of Computers and Directory Structure**

The instructions in this document apply to the specific computing environment described below.

**Placeholder** names are used in **bold** throughout this report for certain **computers, servers, and directories**. Section 2 describes and defines these. Appendix G provides a place for you to write down the particular names and directory paths in your environment.

Aspen Version: Aspen Plus Release 10.1

The user works from a Windows NT personal computer named "**UserNT**".

The Aspen Engine host is a Solaris machine named "**enginehost**". Command language UNIX.

The Aspen GUI software and Microsoft Word reside on the user's Windows NT personal computer named "**UserNT**".

Connection from **UserNT** to **enginehost** is done using F-Secure SSH (App. B.4)

This user executes DOS commands required in these instructions from his/her **UserNT** desktop by launching a Command Prompt window from there.

The **/home/homedir** Directory

Each user in our group has a home directory on the system network of servers that includes **enginehost**. That directory is called **/home/homedir**. It is created and maintained by the System Manager. From **UserNT**, you can map **/home/homedir** onto your Explorer file manager window. From there, you can add lots of subfolders to **/home/homedir**, and you can put files in those subfolders. Once mapped, user files are located in subdirectories of **/home/homedir** are also findable from within Word running on the **UserNT**, and from an Aspen simulation running on the **UserNT** and **enginehost**. When using Word or Aspen, files may be Opened from or Save-As'd to subdirectories of **/home/homedir**.

The "**/home/homedir/AspenWorkingDirectory**" Directory

Each user in our group, at least those planning on using User Databanks, should establish a subdirectory of **/home/homedir** that will be Aspen's working directory. I call mine **AspenWorkingDirectory**, and that is how it will be referred to in these instructions. This is where you put .inp, .rep, and some databank .dat files that you are actively working with. This is the directory from which you execute most of the UNIX commands. User Databanks should be filed here. This is also the directory given in the last line of the Aspen log-on dialog box, as the Aspen Working Directory

Engine Directory:

The directory on **enginehost** that contains the Aspen engine files and databank files is: "**/enginedirectory/**". Appendix G gives examples. **/enginedirectory/** includes files such as:

aqu92.dat  
pure10.dat



## In-House and User Databanks, Configuration 1

combust.dat  
inhspcd.dat      [if an In-House databank has been added]  
...

### GUI **custom** Directory:

The GUI software on **UserNT** includes the directory "**\custom\**", which is used in databank work. Appendix G tells how this directory might be named. **\custom\** contains files such as:

recdef.apr  
custom.bkp  
heatx.dat  
tbcustom.dat  
and "u1bui.dat" and/or "ihpui.dat" if you add them.

### **3. General Comments and Strategy on User and In-House Databanks**

Page 0 of Configuring Databanks and p. 1-2 of the Physical Property manual discuss the differences between User and In-House databanks.

For very small databanks having only a few components and parameters that are mostly available at the outset, a good strategy is to create the entire databank at once from one databank input file. Using this approach with a User databank might be useful occasionally within a business group or program if an individual is concerned with a modest number of compounds or properties, the data are not on the In-House databank, and all of the needed data are available at the outset.

More commonly in a business group or program, there will be large databank(s) that evolve over time and several users who will need to use a quality-assured databank(s). In this circumstance one strategy is for new data to be entered and tested in a User databank owned by the "databank engineer" and then be added to master In-House databank(s) that all users have access to. The data can be added a little at a time, making it easier to check for correctness. This strategy is slow but assures quality best. A faster approach, but more prone to errors, is for all the data available to be typed on to one master In-House databank for immediate use by all members of the group or program.

It is very easy to incrementally add new properties and parameter values to databank files, or to change parameter values. It is very easy to add new components to databank files. When adding or changing data, there are two approaches. Either a master dfms input file can be updated and run to redefine the entire databank from scratch, or a smaller dfms input file can be created that contains only the instructions to change the databank.

It is straightforward to modify the Aspen GUI on each users **UserNT** to tell it of the existence of new components as they are added to a User or In-House databank. This is very timeconsuming though, especially if there are many users, so new components should be added in bunches all at once if possible.

Component names and aliases should be chosen carefully the first time they are added to a databank. There is no convenient way of changing their names and aliases later. The choice of component name and alias is important to both the GUI and actual simulation, as discussed later in several places.

Several more specific comments on the mechanics of User and In-House databank modifications are covered in Section 7.2 - 7.5.

## **4. Mechanics of Creating and Modifying Databank Files**

### **4.1 Introduction**

References are Configuring Databanks, p. 1-2 (Steps 1 and 2), the System Management manual, pp. 6- 1-3, and the Physical Property manual pp. 1- 4-16. The latter describes quite well the dfms input language, including its keywords, paragraphs, sentences, and syntax. The numbering of the steps in Sections 4.2 and 5.2 follows that of the Configuring Databanks document.

Creating User and In-House databank files are very similar. They use the same dfms input language in their databank input files, and most of the steps you take to create them are the same. The main differences are where the databank file (**dbname.dat**) ends up, the permissions and commands you need to get it there (Steps 2.A. and 2.B. in Section 4.2), some of the naming conventions (Section 7.1), and syntax of the User Interface Databank Input file.

Commands you put in the dfms input file (Step 2 below) allow you to do any combination of:

- create a databank,
- add one or more components to a databank,
- add one or more properties to a databank,
- tell the databank the numerical value(s) of one or more parameters for one or more components and one or more properties
- copy components, properties, and/or parameter values to a User or In-House databank from a User, In-House, or System databank
- delete components and/or properties from a databank

**A good tutorial would be to follow the step-by-step instructions of Section 4.2 to modify the databank. Then follow the step-by-step instructions of Section 5.2 to interface your GUI to each modified databank. A good set of examples to use when following the Section 4.2 instructions are the seven example databank modifications in Appendix A.4, Part 1. A good set of examples to use when following the Section 5.2 instructions are the seven corresponding example user interface modifications in Appendix A.4, Part 2.**

## 4.2 Step by Step Instructions

### **Step 0. Log on to UserNT and check out directory mappings**

Log on to **UserNT** and work through the instructions of Appendix B.3.

### **Step 1. Create a dfms input file**

**1.A.** Map **enginehost /home/homedir** directory onto your **UserNT "disk" G** (App. B.3.0).

**1.B.** Create a dfms input file in Word. (App.B.7) The Physical Property manual pp. 1- 4-16 and App. A give examples and describe how to use the instructions within the dfms input file to create and modify databank contents as desired. Save the dfms input file with name **dbinname.inp** into the **enginehost** directory **/home/homedir/AspenWorkingDirectory** (App. B.7.3).

**1.C.** Connect to **enginehost** using SSH. Get a prompt from directory **/home/homedir/AspenWorkingDirectory** (App. B.4).

**1.D.** From **enginehost /home/homedir/AspenWorkingDirectory** prompt, use "ll \*.inp" or "ls -la \*.inp" UNIX command (App. D) to make sure that the databank input file **dbinname.inp** is in this directory.

**1.E.** Convert **dbinname.inp** from DOS to UNIX format using command dos2unix. (App. B. 8.)

### **Step 2. Run a dfms run to create or modify the databank file**

**2.A.** For a User databank, skip steps 2.A.1 and 2.A.2, and go to step 2.B.

**2.A.1.** To create or modify an In-House databank, the System Administrator for the **enginehost** computer must give you write permission for files in the Aspen Engine directory (App. B.5.). Check that you can find this directory and that it is the latest version; e.g. /ap101/ and not /ap100/ when version 10.0 has been superceded by version 10.1.

**2.A.2** For creating or modifying an In-House databank, you must set the XDATA environment. (App B.6). This is not necessary (but not harmful) for User databanks. I have not figured out how often you have to renew this step. Sometimes it stays valid for weeks if you keep your SSH session open; sometimes only hours. If it is not current, you will get an error message in Step 2.C; in which case come back and do Step 2.B.

**2.B** Make sure that all databank files that are being modified or copied from in the dfms **dbinname.inp** file that you are about to run (i.e. those databanks that appear in a FILE OLD sentence), are located in either **/home/homedir/AspenWorkingDirectory** (User databank files) or in the **enginedirectory** (where the System and In-House databank files are located).

**2.C** Now, for creating or modifying both In-House and User databanks: from **enginehost** **/home/homedir/AspenWorkingDirectory** prompt, do the following dfms command:

**/home/homedir: dfms dbinname inrpname**

where **dbinname** is the name of the databank input file (without its .inp extension) and **inrpname** is the name you want given to the report file that is generated by this dfms run (without its .rep extension). I recommend making **inrpname** identical to **dbinname**.

Your screen should show messages like: "Begin ...", "PLEASE CHECK REPORT...", "ASPEN PLUS DATAFILE ...", "real ... user ... system..."

dfms reads file **dbinname.inp** and executes the instructions it contains. It creates or modifies the databank and saves it as **dbname.dat**, where **dbname** is the lower case spelling of **DBNAME**. **DBNAME** is the first word after FILE in the input file sentence FILE, and must be one of the Databank Names given in the first column of Table 1.1 on page 1-3 of the Physical Property Data document. dfms also writes a report file, **inrpname.rep** that describes the dfms run.

dfms puts the report file, **inrpname.rep**, into the same **enginehost** directory from which you executed the dfms command (**/home/homedir/AspenWorkingDirectory** as done here). dfms puts User databank files, **dbname.dat**, into the same **enginehost** directory from which you executed the dfms command (**/home/homedir/AspenWorkingDirectory** as done here). dfms puts In-House databank files, **dbname.dat**, into the directory **enginehost /enginedirectory/**.

**2.D.** Check the directory listings (App. D) to make sure these .dat and .rep files got written into the right directories and have "last modified" dates of when you ran the dfms command.

**2.E.** From **UserNT**, open Word. From Word, do File/Open, browse to **enginehost** **/home/homedir/AspenWorkingDirectory** (**UserNT** drive "G"), find the dfms report file named **inrpname.rep**, open it, and read it to see if the dfms run executed without errors and that the databank contents got modified as desired. If there are errors, go back to step 1.B., fix the .inp file, and go through steps 1.C. – 2.E. again.

**2.F.** Now, or at any time in the future, you can check to see what components, aliases, parameters, and parameter element values are contained in this or any databank file. Do steps 1-8 above, and for Step 2, create a report-only dfms input file, as described in App. A.4.2.

**2.G.** Go to Step 3, described in Section 5.2, to tell your GUI about the databank modifications.

## **5. Mechanics of Telling Your User Interface about the Databank File**

### **5.1 Introduction**

Before you can use a User or In-House databank in your Aspen Graphical User Interface, GUI, you have to tell your GUI that the databank exists and where it is. You also have to tell it the name and alias of each component contained in the databank so that you can access this databank component data when running a simulation. It is recommended to tell the GUI additional information about each component, such as component class (p. 6-6 of System Management manual), molecular weight, boiling point, CAS registry number, etc. These additional parameters help you find the component in the databank when using the GUI.

You need to tell your user interface each time a new component is added to a databank. Otherwise it won't know that the compound exists. You do not need to update your user interface if parameters are added or parameter values are changed for existing components.

These instructions follow closely the the instructions given in Steps 3-7 of the Configuring Databanks document and p. 6-4 to 6-8 of the System Management manual. The examples given below and in Appendix A.5 correspond to databank modification examples in Appendix A.4.

### **5.2 Step by Step Instructions**

#### **Step 3. Create a User Interface Databank Input File**

**3.A-C.** Create a User Interface Databank Input file in Word. (App. B.14) The System Management manual pp. 6- 4-6, the Configuring Databanks document Step 3, and Appendix A.5 describe the syntax of these User Interface Databank Input files and give examples. . Section 7.1 discusses how to name the file.

**3.D** Use Explorer to look at the files that are already in **\custom\** to see what I\*.dat and u\*.dat files are already there. In Version 10.1, AspenTech added placeholder files: usrpp1a.dat, usrpp1b.dat, usrpp2a.dat, usrpp2b.dat, usrpp2c.dat; leave these alone and ignore them. If there are any other user interface databank input files there that you are sure are obsolete (say from doing the examples in these instructions), clean house carefully and delete them.

**3.E** Back in Word, Save-As the User Interface Databank Input file with name **uifilename.dat** (see Section 7.1 for guidance on naming this file) as Text Only into **UserNT** directory **\custom\**. (App. B.14.3) Make sure you saved it as .dat and not .txt. Close the file from Word.

**Step 4. Add an INCLUDE line to the GUI's tbcustom.dat file**

**4.A.** From Word on **UserNT**, open file tbcustom.dat that will be found in **UserNT** directory \custom\. Initially, the tbcustom.dat file should include the following lines. Note that the User Interface Databank Input file names are lower case, unlike Aspen's Configuring Databanks document examples.

```
INCLUDE pure10.dat  
INCLUDE pure93.dat
```

**4.B.** In Word, modify the tbcustom.dat file, inserting additional INCLUDE line(s) for each additional User Interface Databank Input file that you want the GUI to use.

For the examples of Appendices A.4.1 and A.5.1, A.4.3 and A.5.3, and A.4.6 and A.5.6, the tbcustom.dat file should be modified to include the following lines:

```
INCLUDE pure10.dat  
INCLUDE u1bui.dat  
INCLUDE pure93.dat
```

For the example of Appendices A.4.7 and A.5.7, the tbcustom.dat file should be modified to include the following lines:

```
INCLUDE pure10.dat  
INCLUDE ihpui.dat  
INCLUDE u1bui.dat  
INCLUDE pure93.dat
```

**4.C.** After the tbcustom.dat file has the right INSERT statements in it, from Word, Save As type Text Only with filename tbcustom.dat into **UserNT** directory \custom\.

**Step 5. Update User Interface Record Definition Files with DOS cmd: "mmcustom mmtbs"**

**5.A.** From UserNT, open a DOS command window and get to directory `\custom\`. (App. B.10).

**5.B.** At the prompt, `c:\...\custom>`, type `"mmcustom mmtbs <rt>"`. The line will look like:  
`c:\...\custom> mmcustom mmtbs`

After hitting return, this command executes. You should expect a response like "Rebuilding ASPEN PLUS Definition File...". Among the many DOS report lines being spit out, should be "Processing file **uifilename.dat**. N components created in databank **DBNAME**".

**5.C.** Minimize the DOS command window. Don't close it because you will use it again in step 7.

**Step 6. Verify the databank GUI installation using simulation "custom.bkp"**

**6.A.** From UserNT's NT Explorer, browse to the UserNT folder `\custom\`, and double click on the file `custom.bkp`. This will start an Aspen simulation.

**6.B.** Log on to Aspen as usual (App. B.11). The flowsheet window for `custom.bkp` will come up.

**6.C.** With the `custom.bkp` flowsheet, go to the Data/Components/Specifications/Databanks sheet. Under the Available Databanks field you should find the User and/or In-House databank **mmdbname's** given in your User Interface Databank Input Files, **uifilename.dat**. For the Appendix A examples 5.6 and 5.7 taken through Step 5, you would expect to find U1B001 and INHSPCD. Move them over to the Selected Databanks field. Move them up to the top of the list of Selected Databanks. Click over to the Components/Specifications/Selection sheet. Click on the Find button. Search for any component from your databank. Make sure all components can be found. Check that the information you supplied about each component in the User Interface Databank Input file is all there and correct.

(Note that the search goes in order that the selected databanks are listed and once a component is found in one of these databanks, the Find routine quits looking. For example, if INORGANIC and PURE10 are selected, in that order, your search for WATER will only show it in INORGANIC even though WATER is also a component in PURE10. So to check out your databank user interfaces, you may want to move your User and In-House databanks to the top of the list of selected databanks.)

**6.D.** Look at the Run/Settings fields. They should show the existence of the **dbname.dat** files. If no path is shown explicitly in front of **dbname.dat**, Aspen expects to find **dbname.dat** in the user's `/home/homedir/AspenWorkingDirectory` directory on **enginehost**.

**6.E.** Close the `custom.bkp` simulation. (Do not save changes.)



**Step 7. Install the Modified Files on your GUI using DOS command "custinst"**

**7.A.** From **UserNT**, open a DOS command prompt window (or use an existing one that is open) and change directories to get to **UserNT** directory **\custom\** (App. B.10 and App.C).

**7.B.** At the prompt, **c:\...\custom>**, type **"custinst"** and hit <return>. The line will look like:  
**c:\...\custom> custinst**

After hitting Return, this command executes. You should expect a response like: "Copying Customized ASPEN PLUS User Interface Record Definition File custom.cnt 1 file(s) copied"

**7.C.** Minimize, click out of, or close the DOS command window.

**Step 8. Check out the databank by using it in a simulation (See Section 6.)**

**8A.** Now that you think the databank parameters were entered right and the GUI knows about the existence of the databank and the components in it, you should test the performance of the databank. Section 6 describes this.

## **6. Testing and Using User and In-House Databanks in Simulations**

Here are notes that will help you verify that simulations can make use of the new User and In-House databanks and that the parameters got entered correctly. These instructions may not be perfect. More experienced Aspen users will be able to improve on these.

### **Step 1. Start Aspen**

Follow instructions in App. B.11.

### **Step 2. Start a simulation**

Create a new simulation and set it up, or open an existing one and modify it as needed. I have had better luck with process simulation than with a property analysis run.

**2.A.** Data/Setup/Specifications: give it a descriptive title

**2.B.** Data/Setup/Specifications: make Input and Output units SI (that way simulation results will be in the same or similar units to databank units)

**2.C.** Create a very simple flowsheet using blocks that do not require more parameters than you have for the components that you want to check out. At the simplest level, something like a Manipulator-Multiplier.

**2.D.** Save-As your flowsheet, giving it a name like testdb.bkp and saving it as .bkp.

### **Step 3. Select your databanks**

**3.A.** Go to Components/Specifications/Databanks.

**3.B.** Move the desired In-House and User databanks from the left list to the right list and move from the right list to the left list unwanted System databanks.

**3.C.** Put the selected databanks in the priority order you want.

**3.D.** If you are using a User databank, confirm that the simulation knows where the .dat file is by going to: Run/Settings. You should expect something like: ppl1a(or b, or ...): **dbname.dat** if it is in your **/home/homedir/AspenWorkingDirectory** directory.

### **Step 4. Specify and give ID's to your components**

**4.A.** Go to Components/Specifications/Selection. Click on the Find button.

**4.B.** For each component of interest, do the following.

4.B.1. Search for the component, making sure it comes from the databank you want.

4.B.2. Select the component and Add it as a component in the simulation. Close Find.

4.B.2. Edit the Component ID for this simulation if desired to what you would like the component to be called in simulation stream specifications, results, tables, etc.

**Step 5. Complete setting up the simulation**

Finish completing the information needed for the simulation, including:

- 5.A.** Streams/.../ put the components you want into the input streams and set flows.
- 5.B.** Properties/Specifications: Set Property Method etc. to something that makes sense for the parameters available from the databank.
- 5.C.** Setup/Simulation options: Perform Heat balance? check or uncheck this depending on the parameters available in the databank and what you want to test or use.
- 5.D.** If required (I think it usually is), look at and bless Properties/Parameters/Binary
- 5.E.** Choose the stream report options you want. For example, go to Setup/Report Options/Streams: Select mass basis and mole basis plus mass fraction and mole fraction.
- 5.F.** Save your flowsheet.

**Step 6. Define one or more Property Sets that include the properties of interest**

This is **optional**. It is one way to look at properties. Step 9. below is another easier way. Note that some Property Sets have already been pre-defined by Aspen and may include properties of interest to you. If you are content with one or a combination of these pre-defined Property Sets, then skip 6.A. and 6.B. and go to 6.C.

- 6.A.** Go to Properties/Prop-Sets
- 6.B.** Define the Property Set(s) you want, so that properties of interest are included
- 6.C.** Go to Setup/Report Options/Stream/PropertySets and add the Property Set(s) that you want to be included in the simulation results stream reports.

**Step 7. Run the flowsheet.**

**Step 8. Look at the simulation results**

- 8.A.** Did the simulation find the components?
- 8.B.** Are the results what you expected?
- 8.C.** If you included Property Sets (step 6) that show stream or component properties, look at the stream reports and see if they are what you expect.

### **Step 9. Use Parameter Results**

Follow these instructions from within an Aspen simulation to see parameters available that are used or available for components in a simulation. I think you have to run the simulation first, but perhaps you can do this after defining components, streams, blocks, reactions, and property method(s).

#### **9.A. View Properties/Parameters/Results**

After completing a simulation run, go to Properties/Parameters/Results to look at values for properties. For example, to check on vapor pressure (PLXANT) property, go to Properties/Parameters/Results/Pure Component/Temp Dependent/PLXANT.

Note: For T-dependent parameters, the results show which databank the values come from. If a component exists in multiple databanks, you can have values for some parameters coming from one databank and values for other parameters coming from other databanks, depending on the ordering of the databanks and what parameter values they contain. The results do not show the databank source of scalar parameters.

#### **9.B. Create Full Parameter Results Report**

For a more detailed, better-documented report on properties that you can print out.

9.A.1. Select Tools/Retrieve Parameter Results.

9.A.2. Click OK on Retrieve Parameter Results form

9.A.3. Click OK for View parameter results now

9.A.4. Select the View menu, click Report

9.A.5. On Report form, select Simulation in the box for "Display report for:" Click OK

9.A.6. The entire report will be displayed on Notepad. One can view or print. (Note that the report is very long. The full printout would be huge.) The title of the table of interest is "Properties Actually Used in the Simulation". I'm not sure that all these properties are actually used - it may be a list of properties available to the simulation (see Appendix F for a question about this).

### **Step 10. Use Tools/Analysis/Property**

**10.A.** Select Tools/Analysis/Property

**10.B.** Select the type of property or properties of interest (/Pure or /??)

**10.C.** Define the property, units, and independent variable range as asked and hit Go

**10.D.** Check the results that are tabulated or plotted to what you calculate by hand for the parameters that you think you put in.

**10.E.** For example, to check on predictions of vapor pressure (to assure that the PLXANT parameters got into the databank right: go to /Pure: and set: Property=PL; Units=N/sqm; temp range from 293.15 to 393.15K; Go. Check results plotted or tabulated to those calc by hand for PLXANT parameters put in.

## **7. Miscellaneous Notes on Creating and Modifying Databank Files**

### **7.1. Names, passwords, file names, extensions, and case for databanks, etc.**

This section lists and defines databank names, passwords, and filenames, discusses requirements for them (including file extensions and UPPER/lower/Title case), and makes recommendations for creating names. Throughout this report, **placeholder** names are used, often in **bold** font, for databank names, passwords, and filenames.

All the names can be confusing, and the definitions, terminology, and examples in the Aspen documentation is not adequate for sorting this out easily. For example, a single User databank will have the following associated with it: a dfms input file prepared as **dbinname.txt** and then Saved-As with name **dbinname.inp**, a dfms report file named **inrpname.rep**, a databank file named **dbname.dat**, a databank name **DBNAME** or **dbname** (case depends on the context), a password **DBPASSWD** (or **pw**), a databank type **Type** as defined on p. 1-3 of the Physical Property Data document, a databank type **dbtype** as defined on p. 6-4 of the System Management documentation, a User Interface Databank Input File prepared as **uifilename.txt** and then Saved-As with name **uifilename.dat**, and a databank name, **mmdbname**, that is displayed on the GUI's DATABANK sheet of a Simulation. There are constraints on the naming of some of these items, but not on others. Some of the files must be created within and/or placed into specific directories on specific computers.

Aspen, dfms, UNIX, DOS, and occasionally NT are often very fussy with respect to case (UPPER, lower, Title), and file extension (\*.txt, \*.doc, \*.inp, \*.dat, etc.). The Aspen documentation is insufficiently fussy, at least for our computing environment. Follow the conventions prescribed here. I have tried to be precise.

#### **7.1.1 Databank name, **DBNAME****

The databank name, **DBNAME**, must be one of those listed in the first column of Table 1.1 on p. 1-3 of the Physical Property manual. **DBNAME** is the first word after FILE in the dfms input file sentence FILE. **DBNAME** is in CAPS (upper case) and has no file extension when referring to the databank name within dfms input files, in most GUI usages, and in Aspen simulations. For example, in A.4.1, **DBNAME** = USRPP1B.

#### **7.1.2 Databank file name, **dbname.dat****

The databank file name, **dbname.dat** must be identical to **DBNAME**, except it is lower case and it must be followed by file extension ".dat". Sometimes **dbname**, without the .dat extension written, is used to refer to "the name of the databank file". For example in A.4.1, the dfms command sets **dbname** to usrpp1b.dat.

### 7.1.3 Databank password, **DBPASSWD** (sometimes abbreviated **pw**)

The password, **DBPASSWD**, for all **In-House** (and all **System**) databanks is required to be the databank name itself, **DBNAME**, in CAPS, with no extension. I think there can only be the four specific In-House databank passwords listed in the Physical Property manual and no more, namely **INHSPCD**, **INHSSOL**, **IHSAQUS**, OR **INHSBIN**.

The password, **DBPASSWD**, for **User** databanks may be any user-chosen password, eight characters or less, in CAPS, with no extension. I suggest **U1AnnnZZ**, **U1BnnnZZ**, **U2AnnnZZ**, and **U2BnnnZZ**. The first three characters indicate unambiguously the databank name. The next three characters are a serial number, **nnn**, and the last two characters would be the user's initials, **ZZ** in general and **DC** in my case. For example, in A.4.1, we set **DBPASSWD** = **U1B001DC**. Having different passwords for databanks that have the same name and filename allow different databanks to be moved in and out of the appropriate folders and referenced separately in the user interface. I don't understand this thoroughly and list it as an issue in Appendix F.3.

### 7.1.4 dfms databank input files, **dbinname.inp**, and report files, **inrpname.rep**

The name of the input file used by dfms to modify a databank file, **dbinname.inp**, must be lower case, despite Aspen examples and documentation that suggest otherwise. The name **dbinname** may be chosen by the user, but must(?) be eight characters or less, excluding the extension. I suggest using the first three characters to indicate unambiguously the name of the databank being modified. From Physical Property Data manual p. 1-3, I suggest: **ihp**, **ihs**, **iha**, **ihb**, **u1a**, **u1b**, **u2a**, and **u2b**. This leaves five characters to convey additional information, such as the user's initials, and/or a serial number. Preliminary versions of dfms input files that are prepared in Word (or another text editor of your choice) may be saved in the user's normal data folders on **UserNT** as text files named **dbinname.txt** before finally saving-as or copying the final version onto **enginehost /home/homedir/AspenWorkingDirectory** as file name **dbinname.inp**.

The report file generated by dfms during a dfms run is given the name, **inrpname.rep**, lower case, where **inrpname** may be chosen by the user. I recommend always having **inrpname** exactly match **dbinname**, so that it is obvious which report file goes with which input file. For example, in A.4.1, the dfms command line (Section 4, Step 2.C.) should read "dfms u1bdca41 u1bdca41" so that dfms will save the report file as u1bdca41.rep .

### 7.1.5 User Interface Databank Input file names, **uifilename.dat**

The User Interface Databank Input file name placeholder is **uifilename.dat**. It must be eight characters or less, not counting the extension. I recommend the following names for **uifilename**: **ihpui.dat**, **ihsui.dat**, **ihau.dat**, and **ihbui.dat** for In-House databanks, and **u1annnui.dat**, **u1bnnnui.dat**, **u2annnui.dat**, and **u2bnnnui.dat**, for User databanks, where **nnn** is the serial number found in the databank's password, **DBPASSWD**. The first three characters of these names specify the databank name, and the last two characters to show that this is a user interface databank input file. Preliminary versions of User Interface Databank Input files that are prepared in Word (or another text editor of your choice) may be saved in the user's normal data folders as text files named something like **uifilename.txt**. When it is all ready to go and be Saved-as into the **UserNT \custom\** folder, save it as Text Only, named **uifilename.dat** .

#### 7.1.6 Databank name displayed on the Databanks sheet of a simulation, **mmdbname**

The name of the In-House or User databank that will be displayed on the Data/Components/Specifications/Databanks sheet of an Aspen simulation is **mmdbname**, where **mmdbname** is defined on page 6-4 of the System Management document as the variable that follows "DBANK ADD " on the first line of the User Interface Databank Input File, **uifilename**. For In-House databanks, **mmdbname** is required to be **inhspcd**, **inhssol**, **ihhsaqus**, or **inhsbin**, depending on the databank's name. For User databanks, I suggest that **mmdbname** be **u1annn**, **u1bnnn**, **u2annn**, and **u2bnnn**, where the first three characters indicate the databank name and the last three are the serial number in the databank's password, **DBPASSWD**.

#### 7.2 Copying from one Databank to Another

If you want to copy all data about a component from one databank to another, you have to keep the component and alias names the same. Don't be misled by the existence of the NEW-NAMES sentence. It does no good, because there is not a "NEW-ALIAS" sentence. Changing the component name without also changing the component alias is useless and/or confusing. See also Appendix F.7.

#### 7.3 Component Names and Difficulty of Changing Component Names

I recommend that the component name be or contain the proper chemical name, and that the component alias be or start with its chemical formula. This is consistent with Aspen's System databanks and will allow a new user to reliably find the compound. See pages 1- 106-129 of the Physical Property Data manual for how Aspen's PURE10 databank does it. One option to consider is using an uncommon letter, perhaps "Q", as the last character of names and aliases that are being put in on a trial basis.

The only way to change component name is to delete all records of a component and run a dfms input file that contains the new name and new alias and ALL parameter values. See also Appendix F.7. For example, if we want to have a component that has all or most of the same property parameter values as WATER in PURE10, but put it in an In-House or User databank and give it a name of WATERQ and an alias of H2O-Q, then we can not use the COPY command to copy values from PURE10 into our databank. Instead, we must type in every single parameter value into a dfms input file. This is because the COPY command does not have a NEW-ALIAS sentence.

#### **7.4 Limitations on Number of Parameters Allowed in a Databank**

The bottom line here is that we must choose ahead of time which 40 parameters we will want a databank to contain. i.e. which ones we expect to use and which ones we will be willing to live without. It will be inconvenient to use any other parameters that are not on the original list of 40. For reference, PURE10 contains 61 parameters.

Table 1.1 on page 1-3 of the Physical Property Data manual says that the Maximum number of parameters that can be held by an In-House or User pure component databank is 40 (20 for binary databanks). Page 1-6 says that there can be up to 16 elements in a unary (pure-component) parameter. I take this to mean that the one element (value) of the property MW counts as one of the 40 parameters, and the nine elements (values) of the property vector PLXANT counts as another one of the 40 parameters.

This makes it impossible to copy all parameters from System databanks. For example, PURE10 has 61 components. The user and the manager of the In-House databanks must decide which 40 parameters will be used, based on which 40 would be most likely to be used in simulations. If some simulations require different parameters, one option would be to have an alternate In-House databank file available to substitute in that has a different set of parameters.

It is possible that the limit is really 49 instead of 40, but I would not count on it. I became puzzled about the limitation of 40 when I tried to copy all parameters for ethanol from PURE10 onto a User databank that started with only a couple of components and a couple of properties in it. It successfully copied 49 (not 40!) parameters before reporting an error message. The 49<sup>th</sup> parameter was CPSDIP, and all seven of its elements were correctly written to my databank. The 50<sup>th</sup> parameter (based on the order of parameters in PURE10) was CPLDIP, which has seven parameters. The dfms error message said "DATA SUPPLIED IS GREATER THAN NO OF ELEMENTS FOR PROPERTY CPLDIP ..." It also printed identical errors for each of the dozen's of remaining properties in PURE10.

#### **7.5 Documenting the source of a parameter value – the "srcode" parameter**

The PROP-DATA parameter "srcode" provides a way to show where the property values come from. It is like a bibliography reference number. In a separate document I plan to establish a system for this. For now, the system is:

srcode	meaning or source
0	test or trial or fictitious
99	reference number not coded yet, but needs to be; or source not documented



## **Appendix A**

### **Notes and Examples for dfms Input Files (Appendix A, Part 1) and User Interface Databank Input Files (Appendix A, Part 2)**

#### **Part 1. dfms Input Files**

##### **A.1. Summary:**

The Physical Property manual pp. 1- 4-16 describes quite well the dfms input language, including its keywords, paragraphs, sentences, and syntax. By putting the right commands in the dfms input file, you can do any combination of the following:

- create a databank,
- add one or more components to a databank,
- add one or more properties to a databank,
- tell the databank the numerical value(s) of one or more parameters for one or more components and one or more properties
- copy components, properties, and/or parameter values to a User or In-House databank from a User, In-House, or System databank
- delete components or properties from a databank

##### **A.2. Character and line formatting.**

Aspen recommends creating the dfms input files using a fixed-width font such as New Courier.

I don't know how many characters are allowed per line. I think wraparound lines (a sentence that is longer than one line but containing no paragraph return or line-feed) are OK, but I'm not sure. I take a conservative approach and use short lines.

I have been unsuccessful in using the DESCRIPTION command, because dfms does not recognize the endquote. You can easily get by without a DESCRIPTION command, or figure out how to get the quote characters to be recognized by dfms, say by telling WORD to use a special ASCII character set or something.

Comment lines begin with a semi-colon and a space.

In command lines, dfms seems pretty flexible in accepting one or multiple spaces between items, in allowing sub-commands to be on the same line as the main command or in their own subsequent line, etc. I haven't figured out the exact rules on this, but most everything I've tried has worked as I had hoped and guessed that it would.

##### **A.3. Making Multiple Databank Changes in One Input File**

dfms seems to work well in doing multiple things. Commands seem to be executed by priority (discussed in the manual) and by order (where it appears in the file) within a given priority. Most every combination I've tried has worked exactly how I guessed it would. Some of these are shown in examples below. Of course, the conservative approach would be multiple input files in a logical sequence, which is not too hard.

#### A.4. Examples

The following examples illustrate many of the ways you can create and modify User and In-House databanks. A good tutorial exercise would be to run through the seven example databank modifications given in Appendix A.4, following the step-by-step instructions of Section 4.2 to modify the databank, and the step-by-step instructions of Section 5.2 to interface your GUI to each modified databank. The seven examples should be done in sequence.

See Section 4, Step 1.B. for where these dfms input file contents fit in to the overall procedure for modifying databanks, and see Appendix B.7 for details on how to put these dfms instructions into a real **dbinname.inp** file.

##### A.4.1. Create new db, add components, add parameters, enter values using COMP-LIST

Create a new User databank of type PP1 having name **DBNAME** = USRPP1B, file name **dbname.dat** = usrpp1b.dat, password **DBPASSWD** = U1B001DC, using an input file named **dbinname.inp** = ulbdca41.inp. Add to this databank components named WATERQ with alias H2O-Q, and COMPX1 with alias ALIASX1. Add to this databank the following properties: MW, which has one element, ATOMNO, which has 10 elements, and NOATOM, which has 10 elements. Put into this databank values for the molecular weights for both components and the molecular formula for water, using the COMP-LIST form of adding values. Write to a report file an echo of this input file, a directory of this databank that lists the components and parameters present, and a data table of all the parameter values.

```
-----
; Example A.4.1
; filename = ulbdca41.inp
TITLE 'Create new User databank USRPP1B with pw U1B001DC'
FILE USRPP1B U1B001DC NEW
WRFILE U1B001DC
NEW-COMP WATERQ H2O-Q / COMPX1 ALIASX1
NEW-PROP MW 1 / ATOMNO 10 / NOATOM 10
PROP-DATA
  COMP-LIST WATERQ COMPX1
  CVAL MW      0 0 1  18.0 100.0
; note that there is no slash at the end of the above line
; a slash must mean "another sentence or phrase of this para. follows"
  CVAL ATOMNO 0 0 1  1 /
                    2 8 /
                    3 0 /
                    4 0 /
                    5 0 /
                    6 0 /
                    7 0 /
                    8 0 /
                    9 0 /
                   10 0
  CVAL NOATOM 0 0 1  2 /
```

## In-House and User Databanks, Configuration 1

```
2 1 /
3 0 /
4 0 /
5 0 /
6 0 /
7 0 /
8 0 /
9 0 /
10 0
PRINT-DIR U1B001DC
PRINT-DATA U1B001DC
END-INPUT
```

---

### A.4.2. Check the Contents of a Databank File

At any time, you can check to see what components, aliases, parameters, and parameter element values are contained in any databank file. You do a dfms run on an input file that only contains instructions to report on what is in the databank. This will not modify the contents of the databank file, but it will open and resave the file, so its "Modified On" date will change.

To list the contents of User databank file USRPP1B, having password U1B001DC, the following file and dfms command would work.

---

```
; Example A.4.2
; filename = ulbrep42.inp,
TITLE 'Report what is in USRPP1B, pw=U1BDC01, file usrpp1b.dat'
FILE USRPP1B U1B001DC OLD
PRINT-DIR U1B001DC
PRINT-DATA U1B001DC ALL
END-INPUT
```

---

It may be convenient to create the following template and save it as text file dbareptz.txt in a convenient directory. Each time you want to check what is in a databank, go through steps 1-7 of the "Mechanics ..." section of this report. For Step 2, modify the dbarepaa.txt file, changing its filename to correspond to the databank of interest, and replacing **DBNAME**, **DBPASSWD**, and **dbname** with the specifics of the databank of interest.

---

```
; Example A.4.2 template
; filename = dbarepaa.inp,
; where dba is a 3-character abbreviation for the specific databank
; and aa is a serial number, that may be used to specify this file
;
; To use, make the following changes:
; Rename file, changing dba to ____; and aa to ____
; In the commands below, make the following replacements:
; DBNAME to ____; DBPASSWD to ____; dbname.dat to _____.dat
; Save-As filename.inp in enginehost /home/homedir/AspenWorkingDirectory
TITLE 'Report what is in DBNAME, pw=DBPASSWD, file = dbname.dat'
FILE DBNAME DBPASSWD OLD
PRINT-DIR DBPASSWD
PRINT-DATA DBPASSWD ALL
END-INPUT
```



#### A.4.3. Add components, add parameters, enter values to existing databank, copy from databank

Add to User databank USRPP1B (**pw**=U1B001DC) the additional component COMPX2 (ALIASX2), the additional properties: TB (boiling point) and TFP (freezing point). Using the PROP-LIST form of PROP-DATA, put into this databank values for MW, ATOMNO, and NOATOM, TB, AND TFP of COMPX2; and put into this databank values for TB, and TFP of WATERQ, and COMPX1. Write input echo, databank directory, and databank data to the report file.

```
-----
; Example A.4.3
; filename = ulbdca43.inp
TITLE 'Add new comps, props, & values to User db USRPP1B(pw=U1B001DC) '
FILE USRPP1B U1B001DC OLD
WRFILE U1B001DC
NEW-COMP COMPX2 ALIASX2
NEW-PROP TB 1 / TFP 1
PROP-DATA
  PROP-LIST MW 0 / ATOMNO 0 / NOATOM 0 / TB 0 / TFP 0
  PVAL COMPX2 46 / 6 1 8 0 0 0 0 0 0 0 / 2 6 1 0 0 0 0 0
              0 0 / 351 / 159
  PROP-LIST TB 0 / TFP 0
  PVAL WATERQ 373 / 273
  PVAL COMPX1 500 / 350
PRINT-DIR U1B001DC
PRINT-DATA U1B001DC
END-INPUT
-----
```

The PROP-DATA paragraph can contain both PROP-LIST/PVAL groups and COMP-LIST/CVAL groups so that values of specific parameters for specific components can be input in the most convenient way.

#### A.4.4. Create new, empty databank

Create a new, empty In-House databank of type PP1 having name **DBNAME** = INHSPCD, file name **dbname.dat** = inhspcd.dat, password **DBPASSWD** = INHSPCD, using an input file named **dbinname.inp** = ihpdca44.inp.

```
-----
; Example A.4.4
; filename = ihpdca44.inp
TITLE 'Create a new, blank In-House databank INHSPCD with pw INHSPCD'
FILE INHSPCD INHSPCD NEW
END-INPUT
-----
```

#### A.4.5. Copy some parameters of some components from a User to an In-House databank

Copy values of MW, ATOMNO, and NOATOM for COMPX1 and COMPX2 from User databank USRPP1B (**pw=U1B001DC**) to In-House databank INHSPCD (**pw=INHSPCD**).

```
-----
; Example A.4.5
; filename = ihpdca45.inp
TITLE 'Selected data from USRPP1B(pw U1B001DC) add/replace to INHSPCD'
FILE INHSPCD INHSPCD OLD / USRPP1B U1B001DC OLD
WRFILE INHSPCD
COPY U1B001DC
  COMPS=COMPX1 COMPX2
  PROPS=MW ATOMNO NOATOM
PRINT-DIR INHSPCD
PRINT-DATA INHSPCD
END-INPUT
-----
```

#### A.4.5b. Copy some parameters of some components from PURE10 to a In-House databank

Copy values of MW, ATOMNO, NOATOM, and PLXANT for ETHANOL from System databank PURE10 (**pw=PURE10**) to In-House databank INHSPCD (**pw=INHSPCD**).

```
-----
; Example A.4.5b
; filename = ihpdca45b.inp
TITLE 'Selected data from PURE10 add to INHSPCD'
FILE INHSPCD INHSPCD OLD / PURE10 PURE10 OLD
WRFILE INHSPCD
COPY PURE10
  COMPS=C2H6O-2
  PROPS=MW ATOMNO NOATOM PLXANT
PRINT-DIR INHSPCD
PRINT-DATA INHSPCD
END-INPUT
-----
```

#### A.4.6. Delete data from databank and change values of certain parameter elements

Delete component COMPX2 (and all of its parameter element values), delete parameter TFP (for all of the components (WATERQ, COMPX1, COMPX2)), and delete the TB parameter element of only component COMPX1 in User databank USRPP1B (**pw=U1B001DC**). Change the value of MW for COMPX1 to 99 (from 100).

The PROP-DATA paragraph can contain both PROP-LIST/PVAL groups and COMP-LIST/CVAL groups. Values of specific parameters can be input in the most convenient way.

```
-----
; Example A.4.6
; filename = ulbdca46.inp
TITLE 'Delete COMPX2(allprops),TFP(allcomps),TB(COMPX1) from USRPP1B'
FILE USRPP1B U1B001DC OLD
WRFILE U1B001DC
DELETE
  COMPS=COMPX2  PROPS=ALL /
  COMPS=WATERQ COMPX1 COMPX2  PROPS=TFP /
  COMPS=COMPX1  PROPS=TB
; in the 2nd DELETE sentence, can't use COMPS=ALL. Must list all comps.
-----
```

# In-House and User Databanks, Configuration 1

PROP-DATA  
 PROP-LIST MW 0  
 PVAL COMPX1 99  
 PRINT-DIR U1B001DC  
 PRINT-DATA U1B001DC  
 END-INPUT

After completing A.4.1 through A.4.6, the report file from A.4.6 should contain the following.

```

-----
FILE: U1B001DC                                03/23/2000                PAGE    2
Delete COMPX2(allprops),TFP(allcomps),T        from USRPP1B

U1B001DC FILE COMPONENT DIRECTORY                03/23/2000
COMPONENT NAME                                ALIAS NAME
WATERQ                                         H2O-Q
COMPX1                                         ALIASX1
-----

FILE: U1B001DC                                03/23/2000                PAGE    3
Delete COMPX2(allprops),TFP(allcomps),T        from USRPP1B

U1B001DC FILE PROPERTY DIRECTORY                03/23/2000
PROPERTY  NUMBER OF PROPERTY  NUMBER OF PROPERTY  NUMBER OF
ELEMENTS  ELEMENTS            ELEMENTS            ELEMENTS
MW        1      ATOMNO        10     NOATOM        10
TB        1      TFP           1
-----

FILE: U1B001DC                                03/23/2000                PAGE    4
Delete COMPX2(allprops),TFP(allcomps),T        from USRPP1B

COMPONENT: WATERQ                                ALIAS: H2O-Q
NAME  SRCODE  REVDATE      NOE  VALUES
MW    0      03/23/2000    1    0.180000D+02
ATOMNO 0      03/23/2000   10    0.100000D+01,    0.800000D+01,
                                0.000000D+00,    0.000000D+00,
                                0.000000D+00,    0.000000D+00,
                                0.000000D+00,    0.000000D+00,
                                0.000000D+00,    0.000000D+00
NOATOM 0      03/23/2000   10    0.200000D+01,    0.100000D+01,
                                0.000000D+00,    0.000000D+00,
                                0.000000D+00,    0.000000D+00,
                                0.000000D+00,    0.000000D+00,
                                0.000000D+00,    0.000000D+00
TB      0      03/23/2000    1    0.373000D+03
TFP     0      03/23/2000    1    0.100000D+36

COMPONENT: COMPX1                                ALIAS: ALIASX1
NAME  SRCODE  REVDATE      NOE  VALUES
MW    0      03/23/2000    1    0.990000D+02
ATOMNO 0      *****      10    0.100000D+36,    0.100000D+36,
                                0.100000D+36,    0.100000D+36,
                                0.100000D+36,    0.100000D+36,
                                0.100000D+36,    0.100000D+36
NOATOM 0      *****      10    0.100000D+36,    0.100000D+36,

```

## In-House and User Databanks, Configuration 1

				0.100000D+36,	0.100000D+36,
				0.100000D+36,	0.100000D+36,
				0.100000D+36,	0.100000D+36,
				0.100000D+36,	0.100000D+36,
TB	0	03/23/2000	1	0.100000D+36	
TFP	0	03/23/2000	1	0.100000D+36	

```
*****
*   ASPEN PLUS DATA FILE MANAGEMENT SYSTEM ENDS EXECUTION   *
*****
```

From this report, note that the DELETE command deleted COMPX2 completely from the databank, but that it did not completely delete TFP from the databank; instead it set all TFP values to 0.1D+36 (Aspen's null value). It is possible that a CLEAN-UP command would completely delete TFP, but I have not tried it.

### A.4.7. Copy all data from all components from a User to an In-House databank

Copy all remaining components, parameters, and values from User databank USRPP1B (**pw**=U1B001DC) to In-House databank INHSPCD (**pw**=INHSPCD). This adds all of the values in USRPP1B to the existing contents of INHSPCD. It does not completely replace the INHSPCD databank with the USRPP1B contents. Where values in USRPP1B are different than values that existed in INHSPCD, the USRPP1B values are put in. Where INHSPCD has values that are absent in USRPP1B, the INHSPCD values are retained. If you want to make INHSPCD to be identical to the USRPP1B databank, you should first delete everything from the INHSPCD databank before copying all from USRPP1B to INHSPCD.

```
-----
; Example A.4.7
; filename = ihpdca47.inp
TITLE 'All data in USRPP1B addto/replaceover INHSPCD data'
FILE INHSPCD INHSPCD OLD / USRPP1B U1B001DC OLD
WRFILE INHSPCD
COPY U1B001DC COMPS=ALL PROPS=ALL
; where data was missing in INHSPCD, data from U1B001DC is added
; where data was in both, data from U1B001DC replaces what was in INHSPCD
; where data was missing in U1B001DC, it is retained in INHSPCD;
;
PRINT-DIR INHSPCD
PRINT-DATA INHSPCD
END-INPUT
-----
```

After completing A.4.1 through A.4.7, the report file from A.4.7 should look like the following.

```
-----
[reprinting of .inp file is on PAGE 1]
-----
```

FILE: INHSPCD	05/03/2000	PAGE	2
All data in USRPP1B addto/replaceover INHSPCD data			
INHSPCD FILE COMPONENT DIRECTORY		05/03/2000	
COMPONENT NAME		ALIAS NAME	
COMPX1		ALIASX1	



# In-House and User Databanks, Configuration 1

COMPX2  
ETHANOL  
WATERQ

ALIASX2  
C2H6O-2  
H2O-Q

---

FILE: INHSPCD 05/03/2000 PAGE 3  
All data in USRPP1B addto/replaceover INHSPCD data

INHSPCD	FILE	PROPERTY DIRECTORY	05/03/2000
PROPERTY	NUMBER OF	PROPERTY	NUMBER OF
	ELEMENTS		ELEMENTS
MW	1	ATOMNO	10
PLXANT	9	TB	1
		NOATOM	10
		TFP	1

---

FILE: INHSPCD 05/03/2000 PAGE 4  
All data in USRPP1B addto/replaceover INHSPCD data

COMPONENT: COMPX1

ALIAS: ALIASX1

NAME	SRCODE	REVDATE	NOE	VALUES
MW	0	05/03/2000	1	0.990000D+02
ATOMNO	0	05/03/2000	10	0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36
NOATOM	0	05/03/2000	10	0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36
PLXANT	0	*****	9	0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36
TB	0	05/03/2000	1	0.100000D+36
TFP	0	05/03/2000	1	0.100000D+36

COMPONENT: COMPX2

ALIAS: ALIASX2

NAME	SRCODE	REVDATE	NOE	VALUES
MW	0	05/03/2000	1	0.460000D+02
ATOMNO	0	05/03/2000	10	0.600000D+01, 0.800000D+01, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00
NOATOM	0	05/03/2000	10	0.200000D+01, 0.100000D+01, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00
PLXANT	0	*****	9	0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36
TB	0	*****	1	0.100000D+36
TFP	0	*****	1	0.100000D+36

# In-House and User Databanks, Configuration 1

COMPONENT: ETHANOL

ALIAS: C2H6O-2

NAME	SRCODE	REVDATE	NOE	VALUES
MW	1	05/03/2000	1	0.460690D+02
ATOMNO	1	05/03/2000	10	0.600000D+01, 0.100000D+01, 0.800000D+01, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00
NOATOM	1	05/03/2000	10	0.200000D+01, 0.600000D+01, 0.100000D+01, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00
PLXANT	1	05/03/2000	9	0.733040D+02, -0.712230D+04, 0.000000D+00, 0.000000D+00, -0.714240D+01, 0.288530D-05, 0.200000D+01, 0.159050D+03, 0.514000D+03
TB	0	*****	1	0.100000D+36
TFP	0	*****	1	0.100000D+36

COMPONENT: WATERQ

ALIAS: H2O-Q

NAME	SRCODE	REVDATE	NOE	VALUES
MW	0	05/03/2000	1	0.180000D+02
ATOMNO	0	05/03/2000	10	0.100000D+01, 0.800000D+01, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00
NOATOM	0	05/03/2000	10	0.200000D+01, 0.100000D+01, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00, 0.000000D+00
PLXANT	0	*****	9	0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36, 0.100000D+36
TB	0	05/03/2000	1	0.373000D+03
TFP	0	05/03/2000	1	0.100000D+36

\*\*\*\*\*  
 \* ASPEN PLUS DATA FILE MANAGEMENT SYSTEM ENDS EXECUTION \*  
 \*\*\*\*\*

## **Part 2. User Interface Databank Input Files**

### **A.5. User Interface Databank Input Files**

User Interface Databank Input files are used, as described in Section 6 to tell the user's Graphical User Interface (GUI) of the existence of the components in User and In-House databanks, and to provide enough information on those components so that the GUI can help you find the right components when running a simulation.

The name of the User Interface Databank Input file is **uifilename.dat**. My recommended rules for naming **uifilename** are covered in Section 7.1.

The examples that follow correspond to the dfms input file examples of Appendix A.4. You can do each one in turn, or you can do (Steps 1-2 of Section 4.2) all of the A.4 examples and then do (Steps 3-8 of Section 5.2) only Examples A.5.6 and A.5.7. Note that the User Interface Databank Input file names are lower case, unlike the Configuring Databanks document examples.

#### **A.5.1. Tell GUI about databank components added by dfms in Example A.4.1**

Tell GUI about components WATERQ (alias H2O-Q) and COMPX1 (alias ALIASX1) in User databank USRPP1B (**pw** =U1B001DC, **dbname** = usrpp1b.dat) from dfms input file u1bdca41.inp done in example A.4.1. Tell the user interface Components/Specifications/Databank sheet to list this databank as U1B001.

Create the following User Interface Databank Input file (excluding ---- lines above and below)

```
-----
/* Comment lines begin and end with slash-asterisks    */
/* Example A.5.1                                         */
/* file name: u1b001ui.dat                               */
DBANK ADD U1B001
USRPP1 U1B001DC
usrpp1b.dat
H2O-Q WATERQ 0
18.0 0.100000E36 0.100000E36
* Other-inorganics
ALIASX1 COMPX1 0
100.0 0.100000E+36 0.100000E+36
*
-----
```

#### **A.5.2. N/A**

It is not appropriate to modify the GUI after only printing out the contents of a databank that the GUI already knows about. For the tutorial, go to Step 1A of Section 4.2 and Example A.4.3.

#### **A.5.3. Update GUI to include new components added in Example A.4.3**

Update the GUI's information to include all the compounds that are now in User databank USRPP1B (**pw** =U1B001DC, **dbname** = usrpp1b.dat) from dfms input file example A.4.3. This means adding component COMPX2 (alias ALIASX2) to the list of components. We can also provide the GUI with a few more pieces of information about components WATERQ and

## In-House and User Databanks, Configuration 1

COMPX1. Tell the user interface Components/Specifications/Databank sheet to list this databank as U1B001.

Modify the User Interface Databank Input file from A.5.1 to the following. The CAS numbers for water and ethanol are used for components WATERQ and COMPX2, since these components are based upon water and ethanol.

Note that this file must contain the complete list of all components in the databank and not just the latest additions to the databank.

```
-----
/* Example A.5.3                      */
/* file name: ulb001ui.dat            */
DBANK ADD U1B001
USRPP1 U1B001DC
usrpp1b.dat
H2O-Q WATERQ 0
18.0 373 0.100000E36
7732-18-5 Other-inorganics
ALIASX1 COMPX1 0
100.0 500 0.100000E+36
*
ALIASX2 COMPX2 0
46 351 0.100000E+36
64-17-5 n-Alcohols
-----
```

### **A.5.4. Not necessary**

It is not necessary to tell the GUI about an empty In-House databank. For the tutorial, go to Step 1A of Section 4.2 and Example A.4.5.

### **A.5.5. Tell the GUI about In-House databank components of Examples A.4.4 and A.4.5**

Tell the GUI about the In-House databank that was created in Example A.4.4 and modified in Example A.4.5. It is type PP1 having name **DBNAME** = INHSPCD, file name **dbname.dat** = inhspcd.dat, password **DBPASSWD** = INHSPCD.

Create the following User Interface Databank Input file.

```
-----
/* Example A.5.5                      */
/* file name: ihpui.dat                */
DBANK ADD INHSPCD
NONE
SYSTEM
ALIASX1 COMPX1 0
100.0 500 0.100000E+36
*
ALIASX2 COMPX2 0
46 351 0.100000E+36
64-17-5 n-Alcohols
-----
```

**A.5.5b. Skip this. Tell GUI about all the INHSPCD contents in A.5.7 after all changes.**



**A.5.6. Update the GUI to include components added to USRPP1B in example A.4.6**

Update the GUI's information to include the latest components now in User databank USRPP1B (**pw** = U1B001DC, **dbname** = usrpp1b.dat) from dfms input file example A.4.6. Provide the GUI with the information that is available on each component. Tell the user interface Components/Specifications/Databank sheet to list this databank as U1B001.

Modify the User Interface Databank Input file from A.5.3 to the following.

```
-----
/* Example A.5.6                      */
/* file name: ulb001ui.dat           */
DBANK ADD U1B001
USRPP1 U1B001DC
usrpp1b.dat
H2O-Q WATERQ 0
18.0 373 0.100000E36
7732-18-5 Other-inorganics
ALIASX1 COMPX1 0
99 500 0.100000E+36
*
-----
```

**A.5.7. Tell the GUI about In-House components from Examples A.4.4, A.4.5, and A.4.7**

Tell the GUI about all of the components that are now in the In-House databank that was created and modified in examples A.4.4, A.4.5, and A.4.7. It is type PP1 having name **DBNAME** = INHSPCD, file name **dbname.dat** = inhspcd.dat, password **DBPASSWD** = INHSPCD.

Modify the User Interface Databank Input file from A.5.6 to the following.

```
-----
/* Example A.5.7                      */
/* file name: ihpui.dat               */
DBANK ADD INHSPCD
NONE
SYSTEM
H2O-Q WATERQ 0
18.0 373 0.100000E36
7732-18-5 Other-inorganics
ALIASX1 COMPX1 0
99 500 0.100000E+36
*
ALIASX2 COMPX2 0 */
46 351 0.100000E+36
64-17-5 n-Alcohols
C2H6O-2 ETHANOL 0
46 351 0.100000E+36
64-17-5 n-Alcohols
-----
```

## Appendix B.

### Toolkit of Computer Skills for Databank Work

#### B.1. Type the UPPER or lower case accurately

Case matters. Some of the Aspen documentation is not correct. Use the case specified in this report. This is especially true in UNIX, but also is important in parts of .inp files, DOS, etc.

#### B.2. A minimal ability to work in DOS and UNIX is needed

You should be able to get by with the minimal set of DOS and UNIX commands given in Appendices C-D. You may find it convenient or necessary to view files or do some simple editing of them using vi, for which Appendix E gives a minimal set of instructions.

#### B.3. Check that directories are mapped and accessible as needed

**B.3.0** Map your home directory on **enginehost** onto your **UserNT** directory structure as drive "G"(or lowest unused drive letter).

This will allow you can find files and move them around conveniently using Explore, SaveAs, etc. from **UserNT**. From **UserNT**: Start / Programs / Windows NT Explorer / Tools / Map Network Drive : Drive = G: (or lowest unused letter); Path = \\[**enginehost**IPAddress]\home; Connect As: [**enginehost**username]; [check]Reconnect at Logon?; [check]Expand by Default?; OK.

**B.3.1.** Check that your **UserNT** Explorer now shows your home directory on **enginehost**, /home/homedir, plus subdirectories within it, including /home/homedir/AspenWorkingDirectory

**B.3.2.** Check that "Save-As" commands from software programs running on **UserNT** , such as Word and the Aspen simulator, allow you to browse to the **enginehost** directory /home/homedir/AspenWorkingDirectory and save files to it.

**B.3.3.** Check that your **UserNT** Explorer can browse to the \custom\ directory on **UserNT**. Check that applications, such as Word, that run on **UserNT** allow you to browse to the \custom\ directory on **UserNT** and open files from it and save files to it. (**Appendix G** gives the path to \custom\.

B.4. Log into **enginehost** from **UserNT** and open a UNIX command window there.

B.4.1.i. From **UserNT**: Start / Programs / F-Secure SSH / F-Secure SSH : Host Name = **[enginehost.address]**; User Name = **[enginehostusername]**; Auth password = **[enginehostpassword]**. You should get a UNIX window and prompt:

**"enginehost{enginehostusername}n:"**. Type **"pwd"** <rt> to confirm that you are in directory **/home/homedir**. Your home directory, **/home/homedir**, can follow you around from server to server. Type **"cd AspenWorkingDirectory"** <rt> to make **/home/homedir/AspenWorkingDirectory** the present working directory.

B.4.1.ii. An alternate method that I have been told about but have not tried is the following. From **UserNT**: Start/Run. Type in **"\\perle"** and click ok. Login as in B.4.1.i. above. Go to the folder named home.

B.4.2. Tell it that you want your backspace key to act like one (App. D).

B.4.3. To quit out of this, hit the X in the upper R of the window

B.5. Get write permission in the Aspen engine directory

This is needed to install or modify In-House Databanks if System Administrators are not reliably and readily available to do Steps 5-7 of Section 4 for you. Ask one of the System Administrators to give you write privileges to files within the **enginehost** directory: **/enginedirectory/**. You need a permission level that will allow you (and/or the dfms command running from **/home/homedir/AspenWorkingDirectory**) to be able to: cd [to that directory]; ll within that directory; move and copy files to that directory; rename and remove files to and from that directory, and modify files within that directory.

B.6. Set XDATA environment

Need to do this to write to an In-House databank located in the engine directory. From **enginehost** directory **/home/homedir/AspenWorkingDirectory**, type the command: **"setenv XDATA /enginedirectory"** <rt>, where **/enginedirectory** is specified in **Appendix G**.

B.7. To put a text file onto **enginehost** that can be read by dfms, such as a **dbinname.inp** file:

**B.7.1** Open Word

Launch Word from your **UserNT** by selecting Start/Programs/Microsoft Word.



**B.7.2** Type the file within Word

See Appendix A and the Physical Property manual, pp. 1- 4-16 for what dfms input files should contain. To use the dfms input file examples of Appendix A.4, type them into a blank Word file or copy them from an electronic copy of this document and paste them into a blank Word document. Edit them if you want.

It would be a good idea to save interim versions of this file as Text Only (\*.txt), filename "**dbinname.txt**" in one of your NT data folders. When opening such .txt files from Word, you may be asked to choose a file conversion encoding. Choose Plain Text.

**B.7.3** Save the file from Word in a specific way

When the Word file is just how you want it, do a File / Save-As command from within Word. From within the Save-As window, browse to and select directory = "G" (or whatever you mapped **enginehost /home/homedir/AspenWorkingDirectory** to in App. B.3.0) (App. B.3.2). Set type = Text Only (\*.txt). Set filename = **dbinname.inp**. Make sure the file extension that it gets saved as is .inp and not .txt, even though the file type is Text Only. It will ask you "You'll lose features ... Do you really still want to save in this format?". Say yes.

**or B.7.3.b.** Alternative to B.7.3

Alternatively could SaveAs from Word onto one of your normal data folders on **UserNT** (same parameters as above) and then use NT Explorer to drag file **dbinname.inp** over to disk "G", **enginehost /home/homedir/AspenWorkingDirectory** directory (App. B.3.1).

**B.7.4.** Convert the text file from DOS format to UNIX format

As Saved by Word, there are extra carriage returns, etc. that are part of the DOS text format. You need to convert the **dbinname.inp** file to UNIX format on the UNIX machine. Follow the instructions of App. B.8. to use the dos2unix command.

**B.8. Convert text file from DOS format to UNIX format using dos2unix (get rid of extra ^M's)**

Log into the UNIX machine **enginehost** from your **UserNT** (App. B.4). Locate the file to be converted in **enginehost** directory **/home/homedir/AspenWorkingDirectory**. Get a UNIX command prompt in that same directory. At the prompt, **"/home/homedir/AspenWorkingDirectory:"** type **"dos2unix dbinname.inp dbinname.inp"** <rt>. Ignore the error message about keyboard. The file **dbinname.inp** will now be readable by dfms.

**B.9. To open and view a dfms report file, **inrpname.rep**, using Word:**

Word/File/Open/[browse to disk G (mapped **enginehost /home/homedir** directory)]/select **inrpname.rep**

It asks how to convert. Use Plain Text.

B.10. To open a DOS command window on **UserNT** and get to a certain directory:

From **UserNT**: Start / Programs / CommandPrompt

Use > "cd c:\\" to go back to the main c-drive directory.

Use > "dir" to show the directories that are in the current directory.

Use > "cd" to move to another directory. For example, from c:\0MyDocuments>

type "cd \Program Files\AspenTech\Ap101\GUI\**custom**" to move to that directory and get a DOS command prompt there.

To get out of the command prompt window, click X in upper right box of window

B.11. Start up Aspen from your GUI.

**B.11.1.a. Initiate from Explorer browser**

From **UserNT** Explorer, browse to an existing simulation or template (typically named \*.bkp) and double click on it.

**or B.11.1.b. Initiate from Start menu**

From **UserNT**: Start / Programs / AspenTech / AspenPlus 10.0-1 / Aspen Plus User Interface. Select a New Simulation or browse to another existing simulation.

**B.11.2. Engine connection dialog box**

After B.11.1, you should get a dialog box titled "Connect to Engine".

Server type: UNIX host

Nodename: **enginehost.address**

Username: **enginehostusername**

Password: **enginehostpassword**

Work Dirctry **/home/homedir/AspenWorkingDirectory**

See Appendix G. for **enginehost** addresses and username conventions.

B.12. Not Applicable to System Configuration 1.

B.13. Not relevant to System Configuration 1.

B.14. Prepare a text file on Word, such as a **uifilename.dat** and save it onto directory \custom\  
The path to directory \custom is given in Appendices F and B.3.3.

**B.14.1** Open Word.

Follow instructions in B.7.1

**B.14.2** Type the file within Word

See Appendix A.5, the System Management manual pp. 6- 4-6, and the Configuring Databanks document Step 3 for the content and syntax of User Interface Databank Input files. To use the **uifilename** examples of Appendix A.5, type them into a blank Word file or copy them from an electronic copy of this document and paste them into a blank Word document. Edit them if you want.

It would be a good idea to save interim versions of this file as Text Only (\*.txt), filename "**uifilename.txt**" in one of your normal data folders. When opening such .txt files from Word, you may be asked to choose a file conversion encoding. Choose Plain Text.

**B.14.3** Save the file from Word in a specific way

When the Word file is just how you want it, do a File / Save-As command from within Word. From within the Save-As window, browse to and select **UserNT** directory = \custom. (App. B.3.3) Set type = Text Only (\*.txt). Set filename = **uifilename.dat**. Make sure the file extension that it gets saved as is .dat and not .txt, even though the file type is Text Only. It will ask you "You'll lose features ... Do you really still want to save in this format?". Say yes.

**or B.14.3.b. Alternative to B.14.3**

Alternatively could SaveAs from Word onto one of your normal data folders (same parameters as above) and then use your **UserNT** NT Explorer to drag file **uifilename.dat** over to **UserNT** directory \custom. (App. B.3.3)

## **Appendix C.**

### **Some DOS Commands**

Some help on DOS commands can be found in NT help. I forget how to find it.

> dir [dirpath]

lists files and folders contained in the directory specified by dirpath

where [dirpath] is

{blank} (current directory)

dirname

c:\,

c:\dirname\dirname\

> cd [dirpath]

changes the current directory to the directory specified by dirpath

where [dirpath] is something like

c:\,

c:\dirname\dirname\

dirname

\ (goes back up the directory tree)

d: (typed without the cd command will sometimes change your directory to drive d:)

If filenames (or folders) contain spaces or exceed eight characters in length, you are supposed to enclose the path in quotation marks [ I tried this and it did not work. An alternative that seemed to work is to use the first 6 characters followed by ~ followed by 1 {?or another integer?}] An even better alternative seemed to be to naively just type the full filename or foldername, including spaces and long length. This seems to work, perhaps because the NT version of DOS interprets it for me.

> exit

quits the command prompt window (so does clicking the X in upper right of window)

\* works as a wildcard character(s)

## Appendix D.

### Some UNIX Commands

A good reference is UNIX for Dummies, Quick Reference, by Margaret Levine Young and John R. Levine. Use its index to find the commands you need. It is organized alphabetically by command and does a clear job of defining syntax.

All UNIX commands and filenames and file extensions are case sensitive. If a filename has an extension, it must all be specified.

**Make backspace key work** like a backspace key. You will have to redo this each session logged on to the UNIX host computer.

: stty erase [hit backspace key] <rt>

**\* works as a wildcard** character(s)

**To see if a command is available** for execution in the present working directory

: which commandname <rt> For example, : which dfms <rt>

**To see what a command will do**

: man commandname <rt>

: cd [dirpath] <rt> **changes current directory** to directory specified by [dirpath]  
[dirpath] =

/home/homedir goes to your home directory

/enginedirectory (for example, cd /opt/aspens/ap101/aspensplus) goes to the directory where most of the ASPEN engine and databank \*.dat files are located.

~/ or ~/ goes to your home directory

../ or ../ goes to the root directory

/dirname1/dirname2/dirname3 goes to dirname3 when it is in dirname2 which is in dirname1 which is either in the current directory or it is the root directory

dirname4 goes to the directory dirname4 when that is in the current directory

[dirpath] syntax: use only directory name if it is downstream of current directory

start with / if it is upstream or in another branch from current directory

for example: /home/homedir/dirname1/fname.suf

dirname/fname.suf if I'm already in /home/homedir

fname if I'm already in dirname1

: pwd <rt> tells you the present working directory

: ll <rt> **lists the contents of the directory** - the present working directory

: ls -al <rt> same as ll, including in the following specific examples

: ll \*.inp <rt> lists all files in the present working directory with extension .inp

: ll fname.suf <rt> lists filename.suf if it is in the present working directory

: ll /enginedirectory/i\*.dat lists all .dat files in that directory that being with i

**change the name of a file and/or move a file**

: mv [olddirpath]oldfname.suf [newdirpath]newfname.suf <rtn>

**make a copy of a file**

: cp oldfname.suf dupfname.suf <rtn>

**delete a file**

: rm fname.suf <rtn>

**list the contents of a file.** use space bar to scroll down

: more fname.suf <rtn>

To **write the results of a command** (e.g. ll or more) **to a file** named fname, add ">fname" to the end of a command. If fname already exists, >fname will erase the old information and write the new. To append the new data onto existing data contained in fname, add ">>fname" to the end of the command.

To **add a printer**, use commands something like:

: lpr <rtn>

: setenv PRINTER printername <rtn>

: setenv LPDEST printername <rtn>

## Appendix E.

### Basic Text Editing in UNIX vi:

If all goes well, you will not have to use vi. It has and would be needed for troubleshooting possible problems with text formatting, character recognition, dos2unix, etc.

#### To look at and edit a file within vi:

: vi fname.suf        gets you into vi

- > :q!        **gets you out of vi without saving anything** you did
- > :wq        writes (saves) **the changes you made to the file and then quits**
- > w         writes (saves) **the changes you made** to the file
- > [escape key]        **puts me in vi command mode** (use when in insert mode)

#### to move cursor:

Using the window scroll bar and changing the window size are asking for trouble.

Don't use mouse.

arrow keys work pretty well, or use:

up    k

down   j

right   [spacebar]

left    h

down and leftend (return) [returnkey]

pagedown   ctrl-f

page up    ctrl-b

where am I?   ctrl-g

go to top of file        |G    where | is shift backslash, above the Enter key

go to bottom of file     G

To **show line numbers**, type    :set nu

#### to **delete a character**:

position cursor over character and hit the "x" key

#### to get into **insert mode**

position cursor (before or after?) text to be inserted and hit the "i" key

type text to be inserted

hit the [escapekey] to leave insert mode

to **delete a line**: position cursor in line and type dd

to **insert a line**: position cursor in line above and hit the "o" key (puts into insert mode)

hit the [escape key] to leave insert mode



## Appendix F.

### Questions to Resolve

#### F.1. Properties used in simulations

List of "Properties Actually Used in Simulation" does not appear to be such (see Section 6, Step 9). How can we get a real list of properties actually used in this simulation. How can we see what properties for which components are used in which blocks? How can we track how a parameter is used in the simulation?

#### F.2. DESCRIPTION sentence single quotes

When executing dfms on my dfms input files, it does not like my single quotes around the description in the DESCRIPTION command. Error message is something like: "No end quotes; command ignored". How do I get the correct single quote characters into the text file?" [My best guess, not tested, is to a) find out if it is something particular about Word, or about nonUNIX text editors by using Notepad and/or vi to prepare a simple .inp file; and b) if Word is the suspect, test changes in settings of options such as Tools/Autocorrect/Autoformat/SmartQuotes.]

#### F.3. Multiple User databanks

Configuring Databanks, p. 4, and the System Management manual, p. 6-4, suggest that there can be many User databanks, each with a different password, but presumably each having the same approved databank name. I have not yet explored the possibility of having multiple databanks having the same databank name but different passwords, and do not know yet how one would keep them and their files straight. What are the rules and how does it work to have multiple User databanks that have the same databank name, DBANK, and the same databank file name, dbank.dat, but different passwords, **DBPASSWD**.

#### F.4. Using different directories

It would be nice not to have to use **enginehost /home/homedir/AspenWorkingDirectory** as the directory for all dfms work, all dfms input files, all User databank files, and my Aspen simulation working directory. I need to understand directory structures and how to conveniently use dfms, store User databanks, and run and store Aspen simulations using other directories. My early tries at this were unsuccessful.

#### F.5. When copying from PURE10, limit of 40 parameters is exceeded

How can I copy all parameters for a component, say ETHANOL, from PURE10 to a User or In-House databank without bumping into the limit of 40 parameters. For example, a dfms COPY paragraph like: COPY PURE10 COMPS=ETHANOL PROPS=ALL. See Section 7.4.

#### F.6. setenv XDATA needed despite instructions to contrary

Resolve question of confusion regarding when setenv XDATA is required. Page 6-2 of System Management Version 10 says that I "must EITHER have write permission to the ASPEN PLUS system directory for databanks, OR I must set the environment variable or symbol XDATA to point to the working directory where the databanks are to be created.". I had write permission to the **enginehost** directory **/enginedirectory/**, where the databanks are located. (I confirmed this permission by writing, changing, and deleting test files there.) Yet when I do Section 4, Step 2.C. (dfms) without doing Step 2.B. (setenv SCATA) for an In-House databank, I get an error message telling me I need to set environment XDATA.



### F.7. Changing Component Name and Alias

Get Aspen to add one or preferably both of the two suggestions below (i. and ii.) How do I change the name and/or the alias of a component without deleting it and typing in all the parameter values into a new dfms input file. This is also discussed in Section 7.2 and 7.3.

Example 1. I have entered a piece at a time hundreds of parameter values for a component into a User databank. I realize the name is not correct. Right now I am stuck creating a whole new dfms input file from scratch. I'd like to make a little dfms run that says "change the name and leave everything else how it is."

Example 2. I want to add to a User or In-House databank a component named mycomp that I guess has very similar properties to ethanol, except for its molecular formula and weight. I would like to use COPY and then change the component name, alias, MW, ATOMNO, and NOATOM. Right now there is no way to change names. I must type (or cut and paste) each and every one of the hundreds of parameter values from a printout of PURE10 values into a PROP-DATA paragraph of a dfms input file for MYCOMP.

What is needed is one or both of the following two things.

i.) Have a dfms CHANGE-COMPNAME paragraph that has OLDNAME=\_\_\_ , NEWNAME=\_\_\_ , OLDALIAS=\_\_\_ , and NEWALIAS=\_\_\_ sentences.

ii.) in addition to the NEW-NAME sentence in the COPY paragraph, add a NEW-ALIAS sentence.

### F.8. Overlapping component names and databank contents

It is unclear how the GUI and the simulation work when more than one databank has entries for a given component. Also when aliases are the same but names are different (or vice versa).

The best example for why we care about this is the following. Suppose PURE10 has some parameters for a specific compound. But our User databank has values for other parameters for this compound. We want to use PURE10 parameter values wherever they exist, and our User databank values for that compound wherever PURE10 does not have a value. Right now, I don't see how we can do this. The only way would be to copy all PURE10 values onto our User databank.

I remember previously (Fall '99 or Spring '00, Version 10.0) having either been told or concluded that a) the GUI sees (on a Comp/Spec/Selec/Find) that the component appears in both databanks and gives the user the choice of which to select as a component in the simulation; and b) in the simulation, Aspen gets as many parameters as are available from the higher-ranked databank, then gets any unfilled parameters from the lower ranked databank. Also, if component aliases were the same in two databanks but names were the same, that the rule was: look for name or alias in the first databank and obtain all parameters that are available from it; then look for name or alias in the second databank and obtain all additional parameters that were not supplied by the first databank.

Now (Fall '00, Version 10.1) the databank GUI does not work as I expected it to when a component exists in more than one databank (e.g. WATER is in PURE10 and also INORGANIC databanks; and from examples A.4.6, A.5.6, A.4.7, A.5.7 WATERQ is in both USRPP1B and INHSPCD databanks). When running an Aspen simulation using your GUI, searching for such a component (using Data/Components/Specifications/Databanks/Find as described in Step 5.C. of Section 5.2) only shows the component in the first of the databanks, as prioritized in the Selected Databanks list. It gives no hint that the component (and perhaps additional properties) appears in lower priority databanks.

But after running examples A.4.1-7 and A.5.1-7, I see different behaviors and the differences led to my confusion. USRPP1B and INHSPCD both contain COMPX1 (alias ALIASX1). But when using the simulation GUI (in the simulation custom.bkp) to Find COMPX1 (in Data/Components/Specifications/Selection sheet) it only finds the COMPX1 in the higher-ranked databank, and seems blind to the existence of COMPX1 in the lower-ranked databank. It also only saw one WATER – either the PURE10 or the INORGANIC databanks, whichever was higher priority, but not both.

The "old" rules seem much more useful and flexible.

## Appendix G: Your Particular Computer Names, Directories, Addresses, and Paths

The user or his/her System Manager should fill out the blanks for their specific network, computers, directories, and address names. Protect this information as appropriate for your environment. [brackets] show the form expected for the answers.

Name(s) of **System Administrators** for Aspen and **enginehost** are \_\_\_\_\_  
[names]

**UserNT** refers to your NT machine on which your Aspen GUI is installed and on which you use Word to prepare .inp input files, etc.

The GUI software located on your **UserNT** machine includes the directory "**\custom\**", where many GUI files relating to databank work are located. Examples of **\custom** are:  
c:\Program Files\AspenTech\Apui100\custom , or c:\...\Ap101\GUI\custom ,  
or c:\...\Aspen Plus 10.1-0\Gui\custom , with the exact path depending on the version of Aspen and choices made by the person installing the GUI software.  
**\custom\** = \_\_\_\_\_.

Aspen Engine host computer name: **enginehost** = \_\_\_\_\_  
[abcde]

User's home directory on host computer **enginehost** is: **/home/homedir** = \_\_\_\_\_  
[/home/fghij]

User's Working Directory for Aspen simulations on the Aspen engine host computer **enginehost** is \_\_\_\_\_ **[/home/homedir/AspenWorkingDirectory]**

This is the directory typed in when logging in to Aspen. It is where User databanks go.

Directory on **enginehost** that holds most of the engine files including databank files is something like [/.../opt/aspen/ap101/aspenplus/, depending on where the system manager put it and the version of Aspen (ap101 is version 10.1). **/enginedirectory/** = \_\_\_\_\_.

## In-House and User Databanks, Configuration 1

Aspen Engine host computer IP address: **enginehostIPaddress** = \_\_\_\_\_  
[123.4.567.890]

Aspen Engine host computer test address: **enginehost.address** = \_\_\_\_\_  
[abcde.llnl.gov]

User's username on Aspen Engine host computer: **enginehostusername** = \_\_\_\_\_  
When connecting to **enginehost**, this is the answer to dialog box questions like "Connect As:"  
and "User Name:"

User's password on Aspen Engine host computer: **enginehostpassword** = (don't write down)  
When connecting to **enginehost**, this is the password you use to log on.